CLAIMS

What is claimed is:

1. An M×N packet switch for switching M input packets arriving in each of a sequence of frame times to N output ports, the switch comprising

an input module, having M inputs and B outputs, B > M, for switching the M input packets to M of the B outputs to produce M switched packets during each of the frame times,

a packet buffer including B registers, coupled to the input module, for storing the M switched packets into M available registers during each of the frame times to produce M stored packets, and

an output module, having B inputs and N outputs coupled to the packet buffer, for transferring up to N packets from occupied registers in each of the frame times to the output ports based upon destination addresses contained within each of the stored packets.

2. The packet switch as recited in claim 1 wherein the input module is an M×B crossbar switch, and the output module is a B×N crossbar switch.

3. The packet switch as recited in claim 1 wherein the packet buffer is a one-stop shared buffer memory.

4. The packet switch as recited in claim 1further including queues and their identifiers to store the destination addresses and wherein the output module transfers N_1 packets from the occupied registers in each of the frame times to N_2 output ports indicated by identifiers of the queues, $N_1 \leq N_2 \leq N$.

5. The packet switch as recited in claim 1 further including a register selector for assigning the M of the B registers during each of the frame times to generate M assigned registers.

6. The packet switch as recited in claim 5 further including M header hoppers, coupled to the input module, for storing header information from each of the M input packets in each of the frame times and M addresses of the M assigned registers for the M input packets in each of the frame times.

7. The packet switch as recited in claim 6 further including N queues for storing the addresses of the assigned registers in each of the frame times as transmitted to the N queues from the M header hoppers based upon destination information in the header information of the packets.

8. The packet switch as recited in claim 7 wherein the header hoppers, the register selector, and the queues are coupled via a multi-user bus.

9. The packet switch as recited in claim 1 wherein each of the B registers is a circular shift register.

10. An M×N packet switch for switching M input packets arriving in each of a sequence of frame times to N output ports, the switch comprising

a M×B input crossbar switch, B > M, for switching the M input packets to M of the B outputs to produce M switched packets during each of the frame times,

a one-stop shared buffer memory, including B registers, coupled to the input crossbar switch, for storing the M switched packets into M available registers during each of the frame times to produce M stored packets,

a B×N output crossbar switch coupled to the packet buffer, for transferring up to N packets from occupied registers in each of the frame times to the output ports based upon destination addresses,

a register selector for assigning the M of the B registers during each of the frame times to generate M assigned registers,

M header hoppers, coupled to the input crossbar switch, for storing header information from each of the M input packets in each of the frame times and M addresses of the M assigned registers for the M input packets in each of the frame times, and

N queues for storing the addresses of the assigned registers in each of the frame times as transmitted to the N queues from the M header hoppers based upon destination information in the header information.

11. The packet switch as recited in claim 10 wherein the header hoppers, the register selector, and the queues are coupled via a multi-user bus.

12. An M×N packet switch for switching M input packets arriving in each of a sequence of frame times to N output ports, the switch comprising

input means, having M inputs and B outputs, B > M, for switching the M input packets to M of the B outputs to produce M switched packets during each of the frame times,

storage means, including B registers, coupled to the input module, for storing the M switched packets into M available registers during each of the frame times to produce M stored packets, and

output means, having B inputs and N outputs coupled to the packet buffer, for transferring up to N packets from occupied registers in each of the frame times to the output ports based upon destination addresses contained within each of the stored packets.

13. The packet switch as recited in claim 12 wherein the input means is an M×B crossbar switch, and the output means is a B×N crossbar switch.

14. The packet switch as recited in claim 12 wherein the storage means is a one-stop shared buffer memory.

15. The packet switch as recited in claim 12 further including queues and their identifiers to store the destination addresses and wherein the output means transfers N_1 packets from the occupied registers in each of the frame times to N_2 output ports indicated by identifiers of the queues, $N_1 \leq N_2 \leq N$.

16. The packet switch as recited in claim 12 further including a register selector for assigning the M of the B registers during each of the frame times to generate M assigned registers.

17. The packet switch as recited in claim 16 further including M header hoppers, coupled to the input means, for storing header information from each of the M input packets in each of the frame times and M addresses of the M assigned registers for the M input packets in each of the frame times.

18. The packet switch as recited in claim 17 further including N queues for storing the addresses of the assigned registers in each of the frame times as transmitted to the N queues from the M header hoppers based upon destination information in the header information of the packets.

19. The packet switch as recited in claim 18 wherein the header hoppers, the register selector, and the queues are coupled via a multi-user bus.

20. The packet switch as recited in claim 12 wherein each of the B registers is a circular shift register.

21. A method for switching M input packets arriving in each of a sequence of frame times to N output ports using an M×N packet switch, the method comprising

switching the M input packets to M of the B outputs to produce M switched packets during each of the frame times, B > M,

storing the M switched packets into M of B registers during each of the frame times to produce M stored packets, and

transferring up to N packets from up to N of the B registers in each of the frame times to the output ports based upon destination information.

22. The method as recited in claim 21 wherein the transferring includes transferring N_1 packets from the B registers in each of the frame times to N_2 output ports indicated by identifiers of queues, $N_1 \leq N_2 \leq N$.

23. A method for switching M input packets arriving in each of a sequence of frame times to N output ports using an M×N packet switch, the method comprising

prior to the arrival of the M input packets in each one of the frame times, selecting M available registers in a packet buffer having B registers, B > M, to store the M input packets arriving in the next one of the frame times,

setting up connections in an input module to switch the M input packets to the M available registers,

transmitting the register addresses of the M available registers to header hoppers,

delivering and storing the M input packets to the M available registers using the connections of the input module,

sending headers from the M input packets to the header hoppers,

transmitting the register addresses from the headers of the M input packets to N queues corresponding to destination addresses in the headers of the M input packets,

updating the queues based on the header information provided by the header hoppers,

sending control information to a register selector to inform the register selector of the destination addresses of the M input packets in each of the frame times,

selecting up to N stored packets from the packet buffer for each of the destination addresses based on contents of the queues,

transmitting the up to N selected stored packets to the outputs, updating the register selector to account for any remaining destination addresses for each stored packet, and

transmitting any remaining stored packets to the N outputs in subsequent one or more subsequent frames to clear the remaining stored packets.

24. A method for switching M input packets arriving in each of a sequence of frame times to N output ports using an M×N packet switch, the method comprising the steps of

switching the M input packets to M of the B outputs to produce M switched packets during each of the frame times, B > M,

storing the M switched packets into M of B registers during each of the frame times to produce M stored packets, and

transferring up to N packets from up to N of the B registers in each of the frame times to the output ports based upon destination information.